

### **REMARKS**

Claims 1, 3, 4, 8-10, 12, 13, 15-18, 20, 23 and 33-35 are pending. The specification is objected to, claims 1, 3, 4, 8-10, 12, 13, 15-18, 20, 23 and 33-35 stand rejected under 35 USC § 112, and claims 1, 3, 4, 9, 10, 12, 15, 17, 18, 20, and 33-35 stand rejected under 35 USC § 103. Applicant respectfully traverses the rejections in light of the amendments and the following remarks.

#### **Applicant requests interview**

Applicant respectfully requests an interview if it would expedite disposition of the application. The undersigned attorney would welcome and encourage a telephone conference with Examiner at (210) 601-9795.

#### **Specification objections for antecedent basis**

The rejection objects to the specification for failing to provide proper antecedent basis. Applicant respectfully requests that the objections to the specification be withdrawn and presents the following paragraphs (emphasis added) from the specification which provide antecedent basis for “generating, by the server, emails in an order of generation based upon the groups of recipients” as claimed in claims 1, 10, and 18 and as further limited in claims 33-35.

**[0016]** After recipients are associated with each instance of the private content, different emails are generated for each recipient or group of recipients associated with a distinct instance or combination of instances of the private content, advantageously allowing the user to compose a single email for all the recipients. Some embodiments generate separate emails from the email message for the recipients via an email program on the client computer system. **Other embodiments incorporate functionality into an email server that allows the email server to detect instances of private content associated with specified recipients and general content associated with all of the recipients. The email server can then route versions of the email message to the recipients, redacting emails to exclude instances of the private content from emails for unintended recipients.**

**[0025]** SMTP server 130 may receive the email message from a user, generate emails for each recipient, and route the emails to the recipients. More specifically, SMTP server 130 includes email generator 132 to generate emails for each recipient of a message based upon identified private content and recipients associated with the private content. For instance, email generator 332 may receive an email message from a user including data in a header, footer, or in the body of the message that indicates less than all of the recipients are to receive the private content. Email generator 332 may then generate emails for each recipient, excluding private content that is not associated with the corresponding recipients. In some embodiments, when the message is to be delivered in hypertext markup language (HTML) format, HTML code is included in the emails to prevent the corresponding private content from being displayed to recipients that are not associated with the private content. In further embodiments, copies of the email message may be generated for each recipient and the copies are redacted to remove private content that is not associated with a recipient. For example, text of a message that is not identified as private content may be included in each email generated, whereas, text identified as private content may only be included in emails generated for recipients associated with the private content.

**[0026]** SMTP server 130 may then route the generated emails to the recipients 160 and 170 via IMAP server 140 and POP server 150, respectively. More specifically, IMAP server 140 may provide email service to recipient 160 and POP server 150 may provide email service to recipient 170. IMAP server 140 may receive the email for recipient 160 and store the email in a mailbox for recipient 160, message queue 145. Message queue 145 may be, e.g., a hard drive that stores the email until recipient 160 removes the email from message queue 145. Recipient 160 may communicate with LAN/WAN 120 to access IMAP server 140, review email subjects to decide which emails should be downloaded, and download and/or delete the email from IMAP server 140. Similarly, POP server 150 may receive the email for recipient 170, and store the email in message queue 155. When recipient 170 logs into POP server 150, POP server 150 may begin transmitting the email to recipient 170.

**[0027]** Recipients 160 and 170 may include computer systems similar to client 110 having email readers adapted to retrieve emails from servers 140 and 150, and to display the emails to a user. In many embodiments, users may select and adapt any computer system having an email reader or other email program to be recipients 160 and 170.

**[0028]** FIG 4 depicts an embodiment of a device 400 to employ private email content in email messages. **Device 400 may be integrated with an email program and/or software on an email server.** Device 400 includes hardware and software adapted to interact with a user to compose a single email message to generate multiple emails that include different content depending upon the recipient to which the email is routed. For example, device 400 may include functionality implemented via a pull-down menu, a command button, a pop-up menu responsive to a mouse button, or the like, that allows a user to identify an instance of private content in an email message during and/or after the email message is composed and to associate the instance with one or more recipient(s). The user is typically given the option to associate a single instance of private content with one or more of the recipients designated in the email header.

**[0029]** Device 400 may include a content identifier 420, a content associator 430, and an email generator 440. Content identifier 420 may receive a message 410 and identify one or more portions of the message as private content based upon input from the user. In particular, the user may integrate one or more characters or commands into the text of message 410 while composing the message and, when device 400 is integrated with an email server, for instance, integrate an association between the private content and one or more recipients. In further embodiments, the user may interact with content identifier 420 to identify private content in message 410.

**[0030]** Content identifier 420 may include mark identifier 422 and/or command interface 424. Mark identifier 422 may parse message 410 to identify instances of private content. In some embodiments, mark identifier 422 may search text for characters indicative of private content. In further embodiments, mark identifier 422 may recognize one or more typical formats for messages that include private content and identify the private content based upon the formatting. For example, when device 400 is integrated with an email server, message 410 may include a table that describes locations of private content within message 410. More specifically, the table may include data such as line numbers defining locations of private text in message 410, special characters that demarcate the beginning and end of the private content, and the like.

**[0038]** Once email messages are prepared for each recipient, email generator 440 may format the messages and recipients into a standard email format and route the emails to the recipients via an outgoing email message queue, or output queue. The emails may then be packetized for transmission across a network, transmitted to

**another server, reassembled by the other server, and stored in mailboxes for the corresponding recipients.** For instance, the output queue may store emails in order of receipt and transmit the emails as bandwidth becomes available.

**[0041]** Starting with FIG 6A, the user may type a general message 622 in message window 620 for all the recipients of the email without selecting a particular recipient (element 525) from email header 610. Then, when the user wants to draft a portion of the message specifically directed toward “Joe” 615, the user may click on “Joe” 615 to indicate that an instance of private content (element 520) is about to be typed in for the recipient address represented by “Joe” 615. Indications may then be stored in the email message to identify an instance of private content associated “Joe” 615 (element 535). The indications may be stored in the email message itself or, **if the email message is going to be routed to an email server to generate separate emails for “Joe” 615 and other recipients**, the indications may be stored in a packet that is to accompany the e-mail. Then, the user types text 624 as the instance of private content for “Joe” 615.

**[0045]** After the user composes the email message (element 530), identifies private content, and associates the corresponding instances of private content with both “Joe” 615 and “Anna” 618, the dialogue offers the user the option to preview the content for each recipient. By selecting a recipient in the email header 610, the user can preview the resulting content in an email message for that recipient. For example, if the user selects “Group” from email header 610, the message window 620 may only display the portion(s) of the message that will be forwarded to all the recipients that are represented by “Group”. The email message may be routed to an email server to generate and route emails to each recipient based upon the identification of private content and the associated recipient(s).

**Claim rejections under 35 USC § 112**

Claims 1, 3, 4, 8-10, 12, 13, 15-18, 20, 23 and 33-35 stand rejected under 35 USC § 112 for containing subject matter not described in the specification. Applicant respectfully provides the following support from the specification (emphasis added) and requests that the 35 USC § 112 rejections be withdrawn.

[0017] After recipients are associated with each instance of the private content, different emails are generated for each recipient or group of recipients associated with a distinct instance or combination of instances of the private content, advantageously allowing the user to compose a single email for all the recipients. Some embodiments generate separate emails from the email message for the recipients via an email program on the client computer system. **Other embodiments incorporate functionality into an email server that allows the email server to detect instances of private content associated with specified recipients and general content associated with all of the recipients. The email server can then route versions of the email message to the recipients, redacting emails to exclude instances of the private content from emails for unintended recipients.**

[0027] **SMTP server 130 may receive the email message from a user, generate emails for each recipient, and route the emails to the recipients. More specifically, SMTP server 130 includes email generator 132 to generate emails for each recipient of a message based upon identified private content and recipients associated with the private content.** For instance, email generator 332 may receive an email message from a user including data in a header, footer, or in the body of the message that indicates less than all of the recipients are to receive the private content. Email generator 332 may then generate emails for each recipient, excluding private content that is not associated with the corresponding recipients. In some embodiments, when the message is to be delivered in hypertext markup language (HTML) format, HTML code is included in the emails to prevent the corresponding private content from being displayed to recipients that are not associated with the private content. In further embodiments, copies of the email message may be generated for each recipient and the copies are redacted to remove private content that is not associated with a recipient. For example, text of a message that is not identified as private content may be included in each email generated, whereas, text identified as private content may only be included in emails generated for recipients associated with the private content.

[0028] SMTP server 130 may then route the generated emails to the recipients 160 and 170 via IMAP server 140 and POP server 150, respectively. More specifically, IMAP server 140 may provide email service to recipient 160 and POP server 150 may provide email service to recipient 170. IMAP server 140 may receive the email for recipient 160 and store the email in a mailbox for recipient 160, message queue 145. Message queue 145 may be, e.g., a hard drive that stores the email until recipient 160 removes the email from message queue 145. Recipient 160 may communicate with LAN/WAN 120 to access IMAP server 140, review email subjects to decide which emails should be downloaded, and download and/or delete the email from IMAP server 140. Similarly, POP server 150 may receive the email for recipient 170, and store the email in message queue 155. When recipient 170 logs into POP server 150, POP server 150 may begin transmitting the email to recipient 170.

[0031] Recipients 160 and 170 may include computer systems similar to client 110 having email readers adapted to retrieve emails from servers 140 and 150, and to display the emails to a user. In many embodiments, users may select and adapt any computer system having an email reader or other email program to be recipients 160 and 170.

[0032] FIG 4 depicts an embodiment of a device 400 to employ private email content in email messages. **Device 400 may be integrated with an email program and/or software on an email server.** Device 400 includes hardware and software adapted to interact with a user to compose a single email message to generate multiple emails that include different content depending upon the recipient to which the email is routed. For example, device 400 may include functionality implemented via a pull-down menu, a command button, a pop-up menu responsive to a mouse button, or the like, that allows a user to identify an instance of private content in an email message during and/or after the email message is composed and to associate the instance with one or more recipient(s). The user is typically given the option to associate a single instance of private content with one or more of the recipients designated in the email header.

[0033] Device 400 may include a content identifier 420, a content associator 430, and an email generator 440. Content identifier 420 may receive a message 410 and identify one or more portions of the message as private content based upon input from the user. In particular, the user may integrate one or more characters or commands into the text of message 410 while composing the message and, when device 400 is integrated with an email server, for instance, integrate an association between the private

content and one or more recipients. In further embodiments, the user may interact with content identifier 420 to identify private content in message 410.

[0034] Content identifier 420 may include mark identifier 422 and/or command interface 424. Mark identifier 422 may parse message 410 to identify instances of private content. In some embodiments, mark identifier 422 may search text for characters indicative of private content. In further embodiments, mark identifier 422 may recognize one or more typical formats for messages that include private content and identify the private content based upon the formatting. For example, when device 400 is integrated with an email server, message 410 may include a table that describes locations of private content within message 410. More specifically, the table may include data such as line numbers defining locations of private text in message 410, special characters that demarcate the beginning and end of the private content, and the like.

[0039] **Once email messages are prepared for each recipient, email generator 440 may format the messages and recipients into a standard email format and route the emails to the recipients via an outgoing email message queue, or output queue. The emails may then be packetized for transmission across a network, transmitted to another server, reassembled by the other server, and stored in mailboxes for the corresponding recipients.** For instance, the output queue may store emails in order of receipt and transmit the emails as bandwidth becomes available.

[0042] Starting with FIG 6A, the user may type a general message 622 in message window 620 for all the recipients of the email without selecting a particular recipient (element 525) from email header 610. Then, when the user wants to draft a portion of the message specifically directed toward "Joe" 615, the user may click on "Joe" 615 to indicate that an instance of private content (element 520) is about to be typed in for the recipient address represented by "Joe" 615. Indications may then be stored in the email message to identify an instance of private content associated "Joe" 615 (element 535). The indications may be stored in the email message itself or, **if the email message is going to be routed to an email server to generate separate emails for "Joe" 615 and other recipients,** the indications may be stored in a packet that is to accompany the e-mail. Then, the user types text 624 as the instance of private content for "Joe" 615.

[0046] After the user composes the email message (element 530), identifies private content, and associates the corresponding instances of

private content with both “Joe” 615 and “Anna” 618, the dialogue offers the user the option to preview the content for each recipient. By selecting a recipient in the email header 610, the user can preview the resulting content in an email message for that recipient. For example, if the user selects “Group” from email header 610, the message window 620 may only display the portion(s) of the message that will be forwarded to all the recipients that are represented by “Group”. The email message may be routed to an email server to generate and route emails to each recipient based upon the identification of private content and the associated recipient(s).

**Claim rejections under 35 USC § 103(a)**

Claims 1, 3, 4, 9, 10, 12, 15, 17, 18, 20, and 30-35 stand rejected under 35 USC § 103(a) as being unpatentable over Beyda (US Patent No. 6,636,965, hereinafter referred to as “Beyda”) in view of Gilbert (US Patent No. 6,529,942, hereinafter referred to as “Gilbert”) and further in view of Mansour (US Pub. No. 2002/0111995, hereinafter referred to as “Mansour”).

With respect to claim 1, the rejection cites paragraph 8 (emphasis added) of Monsour, and paragraphs 5-7 (emphasis added) are included to provide context:

[0005] In the context of a wirelessly connected HCD, the following advantageous uses come to mind: access to e-mail, access to the Internet, access to calendars and schedules, and collaboration with co-workers. Unfortunately, most HCDs were originally designed to function as personal computer companions or standalone data banks. By shifting the scenario to focus on direct network connectivity, these devices lose the level of processing functionality they originally had when the personal computer provided their interface to the network. **Historically there have been to be two approaches to solving the problem of remote data access: (1) client side processing where the user device (a “fat” client) functions as a small computer; and (2) thin clients that operate in conjunction with server side processing.**

[0006] In order to provide enough functionality to maintain the perceived value of wirelessly connected devices, some solution providers have taken the classic approach of providing the device with more functionality, thus creating a fat client device. For example, some providers add software and features to their platforms and applications to allow end users to connect directly to their email servers, browse web pages, and download and play streaming media files. The result is an effort to create a product that maps to the broadest segment of the market. However, due to practical



technology requirements, vendors are often forced to add more and more resources to the client devices. Faster processors and additional memory not only add cost to the devices, but the additional power requirements call for larger batteries which compromise both the size and weight of the device.

[0007] Three variables that determine practicality to the end user are portability, affordability, and value. Fat client devices, while benefiting from additional functionality, usually suffer a decrease in portability, affordability, product practicality, and mainstream adoption. In addition, a closer look at the functionality actually being delivered by such fat client devices reveals further limitations. For example, although such devices can usually access simple POP3 and IMAP4 email accounts, they may not be sophisticated enough to negotiate corporate firewalls or communicate with proprietary servers (e.g., Microsoft Exchange and Lotus Domino) to access email or PIM data. As a result, corporate end users must maintain separate email accounts for their wireless HCDs and will have no access to corporate server-based PIM data.

[0008] Thin client architectures can be segmented into three typical categories: web interfaces, virtual machines, and thin clients. Of the three, the stateless web interface category seems to be garnering the most attention with the rising popularity of the wireless application protocol (WAP) among wireless carriers and phone manufacturers. However, whether the format is WAP, hypertext markup language (HTML), or any other extensible markup language (XML) derivative, the basic concept remains the same: employ a stateless browser-based user interface to interact with a server-based application that will handle 100% of the **application functionality** and some of the formatting work. The result (at least for WAP browser implementations) is a client that is small and simple enough to fit on a wide range of inexpensive, low-end devices. By moving in this direction, portability and affordability are addressed, while value is derived from powerful server-based applications. However, although this type of architecture offers some practicality to the end user, WAP phones and other WAP-enabled devices are often limited from a user interface standpoint.

According to the rejection, “Mansour shows performing steps at a server based on input from a client (comprising application functionality: see [0008]).” The cited phrase ‘application functionality’ is highlighted above in bold; and identifying pros and cons of fat client devices versus thin client devices within a background discussion neither teaches nor suggests Applicant’s claims for the server performing the identifying and

generating steps or for determining associations in response to identifying instances. Thus, Monsour, in combination with Beyda and Gilbert or any other reference, does not teach or suggest the content of claim 1 nor the content of corresponding claims 18 and 20.

With respect to claim 3, the rejection cites two passages of Gilbert pertaining to 'marks' of a different nature. The first excerpt is accompanied by the sentence preceding it in order to provide context (Gilbert, col. 4, lines 16-21):

At a step 102, the recipient interacts with their station at which they receive the messages to select the icon or other prompt required to review an encrypted comment. For example, in an e-mail system, there may be visual icon that can be selected by the user. In a voice mail system, the system may provide a voice prompt asking a recipient to select a particular key to receive an encrypted comment.

Note that a message recipient may opt to select an icon, a key, or other prompt to receive an encrypted comment. In contrast, Applicant's use of the term 'mark' delineates the 'bounding edge of the instance' of private content.

Continuing, the second excerpt follows (Gilbert, col. 8, lines 1-16, omitting reference numerals):

In this embodiment, all embedded processing codes are placed in brackets. Embedded processing codes comprise identifier codes and text format codes. In alternative embodiments of the invention, other schemes are used to embed the processing codes without departing from the scope of the invention. Included within the processing codes are identifier codes assigned to each recipient receiving a modified message. For example, line 110 in the exemplary electronic mail message 100 presented in FIG. 5 has recipient John identified by the letter "a" placed in brackets, e.g. [a], and Fred identified by the letter "b", which is also placed in brackets, e.g. [b]. When an embedded processing code is found within a message by the software, the software looks for a reference corresponding to a particular recipient which has been provided by the originating user of the message.

Again, note that the connotation differs from Applicant's claim 3 "wherein identifying the instance comprises identifying a mark adjacent to the instance, wherein the mark is

indicative of a bounding edge of the instance,” and thus, the cited passages of Gilbert do not teach or suggest the content of claim 3 nor the content of corresponding claims 12, 18 and 20.

Regarding claim 4, the rejection utilizes an excerpt from Gilbert (col. 6, lines 38, 47). However, describing highlighting in detail with the following wording “the originating user holds down a left mouse button on a mouse serving as a pointing device and drags the screen cursor serving as a pointing device until the desired text has been highlighted or selected” does not teach or suggest Applicant’s claim 4 either alone or in combination with Meyda or Mansour or any other references.

Concerning claims 9 and 17, the rejection refers to two passages in Beyda (col. 3, lines 3-7 and 53-55, omitting steps):

The present invention provides the ability to create customized messages for a recipient by allowing a user to create a single message containing a message for general distribution and comments that can only be received by selected individuals.

If the answer to [a] step is no, then the messaging system forwards only the common portion of the message to the recipient.

Note that immediately preceding and following lines 53-55, Beyda discusses analyzing an address list in a stepwise fashion which differs from Applicant’s claim 9 “wherein redacting the email message to exclude the instance comprises deleting the instance from the email message” and from claim 17 “wherein the content redactor is configured to remove the one or more instances from text of the emails other than the first email routed to each recipient in the first set of the recipients.”

With respect to claim 10, the rejection again cites Mansour, paragraph 8. Also, the rejection again asserts that “Mansour shows performing steps at a server based on input from a client (comprising application functionality: see [0008]),” and the cited phrase ‘application functionality’ is highlighted above in bold. Identifying pros and cons of fat client devices versus thin client devices within a background discussion neither teaches nor suggests Applicant’s independent claim 10:

An apparatus for employing private content in an email message for recipients, the apparatus being part of an email server and comprising:

a computer processor and

memory, the memory comprising computer instructions comprising:

a content identifier to receive the email message from an email client and, in response, to identify one or more instances of private content in the email message;

a content associator to determine an association between at least one instance and a first set of one or more of the recipients and to associate remaining instances, if any, with additional sets of the recipients; and

an email generator to generate the a first email for each recipient in the first set comprising the at least one instance based upon an association with the at least one instance and to redact the at least one instance from the email message to generate emails for the additional sets of the recipients, if any, to display the email message with the at least one instance when routed to the first set of one or more recipients and to exclude the at least one instance when routed to other recipients, wherein the email generator comprises a recipient selector to identify groups of recipients to receive the same or substantially similar email messages, the email generator to generate emails in an order of generation based upon the groups; and

a messaging gateway coupled with the email generator to transmit the emails to the recipients.

Thus, Monsour, in combination with Beyda or any other reference, does not teach or suggest the content of claim 33, 34, and 35.

Regarding claim 15, the rejection cites the following passage from Gilbert (col. 5, lines 6-25, omitting steps and emphasis added):

After text format commands have been embedded within the e-mail message, the software prompts the user to select the recipient(s) who are to receive a modified e-mail message. This may be accomplished by the originating user assigning an identifier code to correlate embedded text commands for a particular recipient's message. One embodiment of implementing an identifier code is for the user to select a number or letter

and place it with the recipient's e-mail name as embedded information at the start of the message. **Embedding an identifier code to a recipient's e-mail name is accomplished similar to embedding text format commands as discussed.**

For example, if Smith is a specific recipient and his e-mail address is Smith@xyz.com, the software allows an originating user to embed [Smith=3] at the start of the message. **Embedded text format commands corresponding to the number "3" thus identify Smith as the specific recipient for the text format changes.** One skilled in the [sic] recognizes that there are **other alternative methods for assigning identifier codes to the recipient.**

This passage discusses embedding identifier codes and embedding text format commands. On the other hand, Applicant's claim 15 covers a feature of the content associator, "wherein the content associator is configured to generate an index associating the first set of the recipients with at least one location of one or more instances in the email message" of the private content. Therefore, Gilbert, either alone or in combination with any other reference, does not teach or suggest the content of claim 15.

With respect to claims 33, 34, and 35, the rejection again cites Mansour, paragraph 8. Also, the rejection again asserts that "Mansour shows performing steps at a server based on input from a client (comprising application functionality: see [0008])," and the cited phrase 'application functionality' is highlighted above in bold. Identifying pros and cons of fat client devices versus thin client devices within a background discussion neither teaches nor suggests Applicant's claims wherein the email server generates the email comprising the email message with the instance and additional emails in the order, wherein the order comprises composing a general message for all recipients, thereafter composing a detailed message for a group of the recipients, and thereafter composing specific messages for individual recipients that are part of the group of the recipients. Thus, Mansour, in combination with Beyda or any other reference, does not teach or suggest the content of claim 33, 34, and 35.

Claims 8, 16 and 23 are rejected by as being unpatentable over Beyda in view of Gilbert and further in view of Mansour and Rafal (US Pub. No. 2002/0002586, hereinafter referred to as "Rafal"). The rejection specifies that the following excerpt from Rafal discloses 'using hypertext markup language code to display email content':

[0037] A party may be recurring; that is, a party may occur weekly, monthly, annually, or at some other period defined by the host. The host may also specify at 110 the time(s) at which invitations are emailed to invited guests as indicated at 117. The invitation may take the form of standard text-only email or may be written in HTML to incorporate "decorations" from the decoration and layout data at 118 described below. In addition, the email message may advantageously include the URL of a party-entry web page which provides information about the party and further acts as the entry point for the party when it occurs, at which time the party entry point Web page will make available a log-in form for use by the invited guest.

Rafal mentions HTML when describing party invitations as either text-only or as embellished with decorations when 'written in HTML' which does not disclose the content of claims 8, 16, and 23:

8. The method of claim 1, wherein generating the one or more additional emails for the one or more remaining recipients comprises generating the email with hypertext markup language code that is configured to display the email message without the instance in response to routing the email to the one or more remaining recipients.
16. The apparatus of claim 10, wherein the email generator comprises a content redactor to generate hypertext markup language code that is configured to display the one or more instances in response to routing the email to a recipient belonging to the first set of the recipients.
23. The computer program product of claim 18, wherein generating emails comprises generating an email having hypertext markup language code that is configured to display the instance in response to routing the email to the recipient.

Therefore, Rafal, in combination with Beyda or Gilbert or any other reference, does not teach or suggest the content of claims 8, 16, and 23.

Claim 13 is rejected by as being unpatentable over Beyda in view of Gilbert and further in view of Mansour and Altavilla (US Pub. No. 2002/0194280, hereinafter

referred to as “Altavilla”). The rejection specifies that the following excerpt from Altavilla discloses ‘the table comprising a set of tags attached to recipients’:

[0025] The process of creating recipient addresses to receive the mail message, and indicating the relative attention level, and any highlighted portions of the message, results in a set of tags as shown in FIG. 3 attached to each address field of each recipient to receive the message. In the example shown, addresses JPL, ABG, PPL, BAA, GRP receive messages having tags identifying different attention levels, and a tag representing portions of the mail message to be highlighted for the recipient. In the example shown, JBL receives the highest attention level (TAG1=1, "Addressee Action Required"), as he is responsible for setting up the meeting which is the subject of the message. PPL receives the message as an FYI message which will be indicated on the respective display monitors of the recipient.

However, attaching tags to email addresses for indicating a degree of priority of a message and highlighted portions does not teach or suggest parsing a table as disclosed by claim 13 for “the apparatus of claim 10, wherein the content identifier is configured to parse a table, wherein the table identifies a location of the one or more instances in the email message.” Hence, Altavilla, either alone or in combination with Beyda or Gilbert or Mansour or any other reference, does not teach or suggest the content of Applicant’s claim 13.

### CONCLUSION

Applicant respectfully addresses the objections and traverses the claim rejections under USC §§ 112 and 103. Accordingly, Applicant believes that this response constitutes a complete response to each of the issues raised in the Office action. In light of the accompanying remarks, Applicant believes that the pending claims are in condition for allowance. Thus, Applicant requests that the rejections be withdrawn, pending claims be allowed, and application advance toward issuance.

No fee is believed due with this paper. However, if any fee is determined to be required, the Office is authorized to charge Deposit Account 09-0447 for any such required fee.

Respectfully submitted,

February 25, 2009

/Dena G Russell/

Date

Dena G Russell, Reg. No. 47,911, Cust. No.: 45670  
Schubert Osterrieder & Nickelson PLLC  
One Congress Pl, 111 Congress Ave, 4<sup>th</sup> fl  
Austin, Texas 78701  
512.692.7297 (tel) 512.301.7301 (fax)  
[dena.russell@sonlaw.com](mailto:dena.russell@sonlaw.com), <http://www.sonlaw.com>  
Attorney for Applicant(s)